

REMARKS

Claims 1-21 remain in this application. Claims 1-3 and 5-21 were rejected by the Examiner under 35 U.S.C. 103(a) as being obvious in view of U.S. Patent No. 6,295,551 to Roberts which discloses downloaded plug-in applets. The applicants submit that the applets disclosed by Roberts are substantially different than the claimed local assistant program. It is well known that plug-in applets are Java programs that can only run within a web browser. Although Roberts does not discuss this fact in detail, it does disclose that each of the plug-in applets require compatible computer operating system and web browsers. (Roberts, Col. 11, Lines 45-51, Col. 12, lines 29-33.) In order for the applets disclosed by Roberts to be active, the web browser program must also be active.

As further evidence of the relationship between applets and browsers, a technical document titled "Applet Caching and Installation in Java Plug-in" from Sun Microsystems is attached. As discussed in the first line of this document, the applets are downloaded by the web browser. Once downloaded, the browser can store the applet in cache memory so that it does not have to be downloaded from the web server every time the user references the applet. As pointed out by the Examiner, this cache storage is described in Roberts. The applicants submit that the applet storage in cache memory is only a means for quickly accessing the program and does not mean that the applet program is continuously active.

The applicants submit that there is a substantial difference between the claimed local assistant program and the applet programs disclosed by Roberts. The applet requires a web browser and if the web browser is closed the applets will also cease to be active. In contrast, the local assistant program is an independent program that does not require a browser program to operate. Because the local assistant program is independent of the web browser program it remains active and continues to process data regardless of whether the browser program is active or terminated. The local assistant program also remains active after the connection to the Web site server is broken.

The Examiner admits that Roberts does not disclose a "ruleset" but argues that it would have been obvious to one of ordinary skill in the art at the time of the invention to modify

Roberts to specify the functionality as a rule set. The Examiner states that Roberts teaches that predefined functionality is requested and downloaded either before or after installation of the applet. The applicants ask the Examiner to understand that the functionality of the applet disclosed by Roberts cannot be modified after the applet has been downloaded.

As discussed in Roberts, the system evaluates the functional needs of the system and then generates an applet that meets these functionality requirements. "After the server 20 evaluates these functions, it then generates the user applet 22 representative of the functions. Once the server 20 generates the user applet 22 representative of these functions, it transmits the user applet 22 to the user computer 12 ..." (Roberts Col. 9, lines 45-51.) Roberts only discloses the downloading of applets with a predefined functionality. If additional functionality is required Roberts discloses that additional applets must be downloaded. "[I]f a user computer 12 enters a session to share a web page as further described hereinafter, the server 20 only downloads functionality to the user computer 12 necessary for the browser 18 to be notified about, and to display the web page. If, for example, the user computer 12 is sharing a software application, then the server 20 will download to the user computer 12 another applet in real time of the next piece of functionality the user computer 12 needs to implement the software application." (Roberts Col. 10, lines 5-13, emphasis added.) Roberts clearly discloses that multiple applets are downloaded based upon the functionality requirements of the computer. The applicants respectfully submit that there is no disclosure or suggestion in Roberts for modifying or altering the functionality of the applets after they have been downloaded to the user computer.

The applicants also submit that in order for the Roberts system to operate, the user computer, the server, and the representative computer have to be continuously connected in an ongoing session. Thus if any of the three computers breaks the network connection, execution of the Roberts applet is terminated. In contrast, the claimed local assistant program functions without the need for a continuous connection to the network and only occasionally requires a connection to the server. while the local assistant continues its background processing whether or not the customer/client computer is connected to a server or to the network.

The claims have been amended to clarify the differences between the local assistant program and the applets disclosed by Roberts. Claims 1 and 8 have been amended to add the limitations that the local assistant program does not require a web browser program, remains active by processing data until disabled or uninstalled regardless of the status of the web browser program or connection between the customer's computer and a server. Claims 2-3 and 5-7 depend from claim 1 and claims 9-21 depend from claim 8. For the reasons discussed above, the applicants submit that claims 1-3 and 5-21 are not obvious under 35 U.S.C. 103(a) in view of Roberts.

With respect to claim 2, The local assistant program is intended to monitor the user interactions with the web browser as well as various other computer programs. In contrast, the applets disclosed by Roberts are integrated into the web browser and only intended to monitor the user's interaction with the computer's web browser. In order to clarify this distinction, the applicants have amended claim 2 to include the limitation that the local assistant program observes, analyzes and/or stores information regarding the customer interaction with the web browser and other computer programs. Although Roberts mentions the computer program Excel, this description is only as an analogous explanation and in no way discloses or suggests that the applet interacts with any program other than the web browser. (Roberts, Col. 9, lines 53-56.) For these reasons and the reasons discussed with respect to claim 1, the applicants submit that claim 2 is not invalid as obvious under 35 103(a) in view of Roberts.

With regards to claim 5, the s also respectfully disagree with the Examiner argument that Roberts discloses "administrative applets" that schedule predefined events. Roberts specifies that the administrative applet performs the specific function of updating the queues information

for call requests from user computers. (Roberts, Col. 15, lines 38-44.) This queues information is based upon user computer actions, not periodic tasks which are performed regardless of the user's computer interaction.

In order to clarify that the system local assistant program performs tasks that are independent of the user's interaction with the computer, claim 5 has been amended to include the limitation that the periodic tasks are scheduled. Because Roberts only discloses a system that responds to the user's interactions with the computer, it does not disclose the performance of any scheduled tasks. For these reasons and the reasons discussed in claim 1, the applicants submit that claim 5 is not invalid as obvious under 35 U.S.C. 103(a) in view of Roberts.

Claim 6 has been amended to add the limitation that the information transmitted to the local assistant program is from multiple web sources. Roberts discloses that all applets are transmitted to the user computer from a single server. Because the system is configured for communications between the server and computer, Roberts does not disclose or suggest receiving information from multiple web sources. For these reasons and the reasons discussed above with respect to claim 1, the applicant submit that claim 6 is not invalid as obvious under 35 U.S.C. 103(a) in view of Roberts.

Claims 10 and 11 include the limitations that the server-side local assistant system includes a merchant database that stores information relating to assisted merchants and provides the local assistant program downloads. In particular, the merchant data base contains information about multiple merchants rather than information about services or products of a single merchant. In contrast to these claim limitations, the applicants respectfully submit that Roberts discloses downloading a host merchant's sales web pages stored on a server that does not

also provide downloads of the local assistant program. Claims 10 and 11 also depend from claim 8. For the reasons discussed above and in claim 8, claims 10 and 11 are not obvious in view of Roberts.

Claims 17 - 21 depend from claim 8 and include all of the limitations of claim 8. As discussed above with respect to claim 8, Roberts discloses downloading applets that include specific types of functionality. Once downloaded, the applet can display certain information through the browser of the user's computer but it does not store any of this information on a database on the user's computer. In contrast, the claimed local assistant program is used with databases to store rules and local interaction data. The local assistant downloads new and updated rules to a rules database and save local interaction data to a local interaction database on the customer computer. In order to clarify this difference, claim 17 has been amended to include the limitation that the rules interpreter reads and writes local interaction data on a local interaction database. Similarly, claim 18 has been amended to add the limitations that the rules interpreter receives rule updates and stores the rule updates on a rules database on the customer's computer system. Claim 19 has been amended to add the limitations that the rules interpreter system receives updated interaction data and stores the updated interaction data on a local interaction database on the customer's computer. For these reasons and the reasons discussed above in claim 8, the applicants submit that claims 17 -19 are not obvious under 35 U.S.C. 103(a) in view of Roberts. The applicants submit that claims 20 and 21 are not obvious under 35 U.S.C. 103(a) in view of Roberts for the same reasons discussed above in claim 8.

Claim 4 was rejected under 103(a) over U.S. Patent No. 6,295,551 to Roberts in view of U.S. Patent No. 6,240,459 to Roberts. Claim 4 depends from claim 1 and has been amended to

include the limitation that the computer mediated customer interaction comprises playing music through a computer program that is independent of the web browser program. As discussed with respect to claim 1, all applet plug-ins are installed on browser programs. The present invention is a program that is functionally independent of the web browser and capable of monitoring user interaction with programs other than the web browser. None of the cited references disclose a system mediated customer interaction is through a computer program that is independent of the web browser program. For these reasons and for the reasons discussed above in claim 1, the applicants submit that claim 4 is not obvious under 35 U.S.C. 103(a) in view of the Roberts reference together with U.S. Patent No. 6,240,459.

Applicants respectfully submit that the claims are patentable over the cited prior art and request that they be allowed. The Examiner is encouraged to call the undersigned collect at (415) 705-6377 if there are any outstanding issues or questions which can be resolved to allow this application to be passed to issue.

Respectfully submitted,
Dergosits & Noah, LLP

Dated: May 18, 2005

By:



Paul K. Tomita
Reg. No. 43,196
Tel.: (415) 705-6377

- ! A JAR file is listed in `archive` but not in `cache_archive`. In this case, the JAR file is cached using the native browser cache. This guarantees a minimum of caching.

`cache_option` and `cache_archive` can only be specified once per `EMBED/OBJECT` tag. In addition, both attributes must be specified. If either `cache_archive` or `cache_option` are missing, Java Plug-in will treat the applet normally using the `archive` attribute.

cache_version

The `cache_version` is an optional attribute contains a list of versions of the files to be cached:

```
<PARAM NAME="cache_version" VALUE="1.2.0.1, 2.1.1.2, 1.1.2.7">
```

Each version number is in the form of X.X.X.X which X could be from 0 to F in hexadecimal. Each version number corresponds to the jar file in the `cache_archive`.

Applet Caching Update Algorithm

By default, without the `cache_version` attribute, applet caching will be updated if:

- The `cache_archive` has not been cached before, or
- The "Last-Modified" value of the `cache_archive` on the web server is newer than the one stored locally in the applet cache, or
- The "Content-Length" of the `cache_archive` on the web server is different than the one stored locally in the applet cache.

However, in some situations, the "Last-Modified" value returned from the web server through HTTP/HTTPS may not reflect the actual version of the applets. For example, if the web server crashes and all the files are restored, the `cache_archive` will have a different modification date on the server. Even if the `cache_archive` has not been updated, it will still force all the Java Plug-in clients to download the `cache_archive` again.

To strongly enforce the version update, we recommend the applet deployer use the `cache_version` attribute. Applet caching will be updated if:

- The `cache_version` corresponding to the `cache_archive` in the `EMBED/OBJECT` tag is larger than the one stored locally in the applet cache.

Unlike the default update, `cache_version` will eliminate the extra connection to the web server to obtain "Modification-Date" and "Content-Length" of the `cache_archive`. This will speed up performance in most cases.

Security

Although sticky applets are cached locally, they will still conform to the security policy defined by its original codebase and signer.

Known Limitations

Both HTTP/HTTPS can be used with applet caching. However, `cache_version` must be used when HTTPS is the protocol because HTTPS in Java Plug-in uses a browser native API, and in some cases the browser native API does not return information like Modification-Date and Content-Length properly. As a result, using HTTPS without `cache_version` may force the `cache_archive` to download each time the applet is loaded.

Caching of the JAR files specified in the manifest's Class-Path variable using Java Plug-in's cache is currently not supported.

JAR files should either be listed in `archive` or `cache_archive`, but not both.

The path specified in the `cache_archive` must be a relative URL to the applet's codebase. Full URLs are not supported in `cache_archive`.

The Source

<http://java.sun.com/products/plugin/1.3/docs/appletcaching.html>



May 11, 2005

Java Plug-in 1.3 Documentation

Applet Caching and Installation in Java Plug-in

Applet caching ensures that applets are not unnecessarily downloaded by a browser every time a user references them.

Java Plug-in has supported disk caching in previous versions by using the same cache the browser uses for all other web documents. This works for small applet usage, but larger applets can often get flushed from the cache to make room for other documents since the browser has no knowledge that they might be needed in the future. The result is that this caching strategy fails where it is needed most in large business applets.

This release introduces an alternative form of applet caching which allows an applet deployer to decide whether an applet should be "sticky", that is, to be stored in a secondary cache which the browser cannot overwrite. The only time "sticky" applets get downloaded after that is when they are updated on the server. Otherwise the applet is always available for quick loading. We recommend that applets which provide core business applications be made "sticky" to improve their startup performance.

This new feature is activated by including the new "cache_option", "cache_archive" and "cache_version" values in the EMBED/OBJECT tags. The use of Java Plug-in is as below:

```
<OBJECT ....>
  <PARAM NAME="archive" VALUE="...">
  ....
  <PARAM NAME="cache_option" VALUE="...">
  <PARAM NAME="cache_archive" VALUE="...">
  <PARAM NAME="cache_version" VALUE="...">
</OBJECT>

or

<EMBED ....
  archive="..."
  ....
  cache_option="..."
  cache_archive="..."
  cache_version="...">
```

cache_option

The cache_option attribute can take one of three values:

- No**
Disable applet installation. Always download the file from the web server without using browser or plug-in cache.
- Browser**
Run applets from the browser cache (default).
- Plugin**
Run applets from the new Java Plug-in cache.

cache_archive

The cache_archive attribute contains a list of the files to be cached:

```
<PARAM NAME="cache_archive" VALUE="a.jar,b.jar,c.jar">
```

Like the archive attribute in the APPLET tag, the list of jar files in the cache_archive attribute do not contain the full URL, but are always down to the codebase specified in the EMBED/OBJECT tag.

Note that the list of JAR files in the cache_archive attribute and those in the archive attribute may be similar but should not contain the same. There are two possible cases: